e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:01/January-2021 Impact Factor- 5.354 www.irjmets.com

VEHICLE DETECTION USING ARTIFICIAL INTELLIGENCE FOR TRAFFIC SURVEILLANCE

Priyank Rawat[*1], Chandan Singh[*2], Ajay Kumar[*3], Ayush Sharma[*4], Mr. Sunny Arora[*5]

[*1,2,3,4] Student, Department of Information Technology, Dr. Akhilesh Das Gupta Das Gupta Institute of Technology & Management, New Delhi, India.

[*5]Assistant Professor, Department of Information Technology, Dr. Akhilesh Das Gupta Das Gupta Institute of Technology & Management, New Delhi, India.

ABSTRACT

The main purpose of the project is to create a system that can monitor the movement of the vehicle. Vehicle tracking applications play a crucial role in traffic management like highway traffic surveillance control and planning of urban traffic. Vehicle tracking processes are used for tracking vehicles, counting vehicles, traffic analysis and categorizing different vehicles and it can be carried out under various environmental changes. Therefore, we can use this model on a traffic light camera to reduce traffic congestion on roads. If we can integrate vehicle detection system in a traffic light camera, we can easily track a number of useful things simultaneously such as the numbers of vehicles that are present at the traffic junction during the day, the time in which the traffic builds up, the type of vehicles that are traversing the junction(heavy vehicles, cars, etc.).This system can help us to reduce the traffic and disperse it through a different street.

Keywords: Vehicle detection, Traffic surveillance, Real time computer vision, Morphological image processing.

I. INTRODUCTION

One of the most important video-based monitoring system is traffic monitoring. Therefore, for years, researchers have been investigating Vision-based Intelligent Transport System (ITS), transport planning and automotive engineering applications to extract useful and accurate information for traffic image analysis and traffic flow control such as car count, car tracking, vehicle tracking, traffic flow, vehicle classification, speed of traffic, changes in traffic line, recognition of license plates, etc. In the past, vehicle detection, tire separation and tracking systems were used to determine various charges for different vehicle types. Recently, a vehicle recognition system has been implemented to detect the lanes and differentiate the type of vehicle class on highways such as cars, motorcycles, vans, buses and more. However this system may not perceive the vehicle properly due to background obstacles such as road signals, trees, weather conditions, etc. The performance of these systems depend on good traffic image analysis methods for detecting, tracking and classifying vehicles. However this system may not perceive the vehicle properly due to background obstacles such as road signals, trees, weather conditions, etc.

II. METHODOLOGY

The proposed vehicle detection system uses video data obtained from stationary street cameras to perform mathematical operations over the set of frames captured from the video to estimate the number of vehicles present at the scene. This system makes use of computer vision to extract the information from the video.

2.1 COMPUTER VISION - Computer vision is a process by which we can understand how photos and videos are stored and how to use and retrieve data from them. Computer Vision is used as the base or most widely used for Artificial Intelligence. Computer-Vision plays a major role in self-driving cars, robots and photo editing applications. Computer vision is implemented using the OpenCV library in python.

2.2 OPENCV - OpenCV is a vast open-source library for the computer vision, machine learning, and image processing and now plays a major role in real-time performance that is very critical in today's systems. By using it, one can process photos and videos to identify objects, faces, or handwriting. When combined with various libraries, such as Numpy, python is able to process OpenCV array structure for analysis. To identify a picture pattern and its various features we use a vector space and perform mathematical functions on these elements. With the help of opencv we will perform various operations on the image obtained by the video .

2.3 IMAGE PROCESSING

Visual information is the most important type of information that is perceived, processed and interpreted by the human mind. Image processing is the process of performing certain tasks on an image, in order to extract more useful information from it. An image is nothing more than a 2-D matrix (3-D in the case of colored images) defined by the mathematical function f (x, y) where the x and y are the horizontal and vertical coordinates respectively. The pixel value defines how bright that pixel is, or what color it should be. In grayscale pictures a single number is used to define the pixel value which represents the brightness of that pixel. The most common pixel format is the byte image, which is stored as an 8-bit number that gives a range of possible values from 0 to 255. As a convention 0 is taken to be black and 255 is taken to be white, the values in between comprises of different shades of gray.

To represent color images, different red, green and blue segments must be specified for each pixel (considering the RGB color model), so the pixel value becomes a vector of three numbers. Usually three different parts are stored as three different `grayscale' images known as color planes (each red, green and blue), which must be recombined when image is displayed or processed. OpenCV gives the flexibility to capture image directly from a pre-recorded video stream, camera input feed, or a directory path which is known as image acquisition.

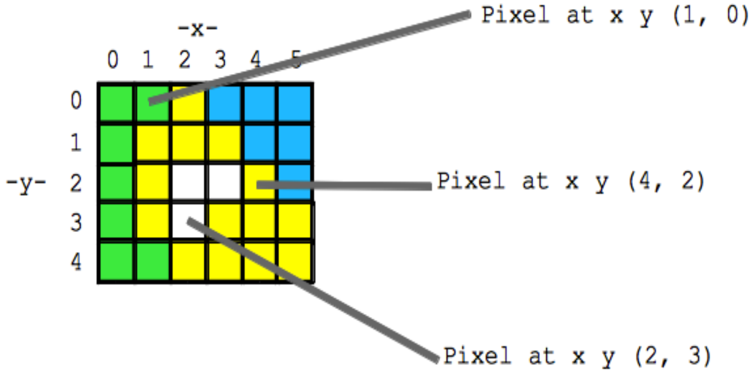


Fig-1: Image representation as a matrix

2.4 BACKGROUND SUBTRACTION

In many vision based applications background subtraction is considered as a major preprocessing step. For instance, think of cases such as a guest counter where a stationary camera captures the number of guests entering or leaving a room or a traffic camera extracting information about the vehicles etc. In all such cases, we first need to remove the person or vehicle. If we have a picture of background alone, like image of the area without visitors, image of the road without vehicles etc then it is an easy job. We just need to subtract the new image from the background to get the foreground objects alone. However in most of the cases, we 'll not have such a picture, thus we want to extract the background from no matter what pictures we've got. It becomes tougher when there's a shadow of car. But in most of the cases, you may not have such an image, so we need to extract the background from whatever images we have. It becomes very sophisticated when the vehicle's shadow is involved. Since the shadow also moves so simple subtraction will mark that also as foreground. In OpenCV we've 3 algorithms to try and do this operation Background Subtractor MOG, BackgroundSubtractorMOG2, Background Subtractor GMG. It is always better to use morphological transformation to remove the noises from the result.

2.5 IMAGE SMOOTHING

As within other signals, images can also contain different types of noise, especially because of the source (camera sensor). Image smoothing techniques helps in reducing the noise. In OpenCV, image smoothing (also called blurring) can be carried out in many ways. In this project we have used the Gaussian filter for image smoothing. The property of Gaussian filters is that there's no overshoot to a step function input while minimizing the increase and fall time. In terms of image processing, any sharp edges in images are smoothed while minimizing an excessive amount of blurring.

2.6 MORPHOLOGICAL OPERATIONS

Morphological operations are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image. We have applied following morphological operations in our project:

1. Dilation - This procedure follows convolution with some kernel of a particular shape sort of a square or a circle. The anchor point of the kernel denotes the center of kernel. To compute the maximum pixel value the kernel is overlapped over the image. After calculating, the image is replaced with anchor at the middle. Through this method, light-colored areas increase in size that is why the image size will increase. For example, the dimensions of an object in white shade or bright shade increases, while the dimensions of an object in black shade or dark shade decreases. In dilation, pixel element is '1' if atleast one pixel under the kernel is '1'. It therefore increases the white region within the image or size of foreground object .It's also helps in joining broken parts of an object.

2. Structuring element - We can manually create structuring elements with the help of Numpy. We can create different shape of kernels such as rectangle, circle etc. The kernel matrix is prepared using the getStructuringElement() method. To get the desired kernel we just need to pass the shape and size of the kernel.

3. Closing - It helps in closing small holes inside the object, or small black points on the object present in the image. It is generally followed by dilation.

2.7 CONTOUR DETECTION

The curves in an image that are joint together are called as contours. The curves join the continuous points that are present in an image. Contours are used to detect the objects. We use find Contours() method to find the contours in the image. We can only see the contours of those vehicles that are present in our detection zone. That's how the vehicles are detected in all the frames.

2.8 TRACKING AND COUNTING

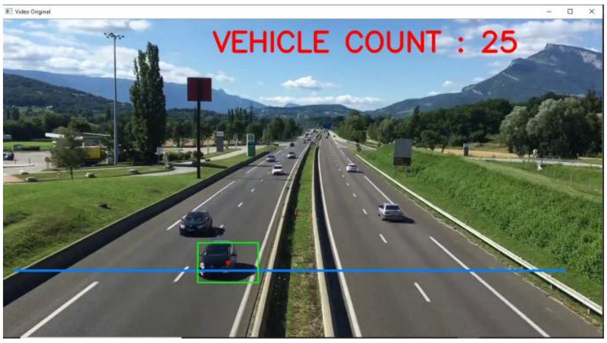
In each frame we use the contour detection to extract the x-coordinate, y-coordinate, width and height of the object. Based on the extracted dimensions of the object, we will loop through them and draw a rectangle around each frame of the image to track it. Counting of vehicle takes place when vehicle crosses the blue line defined by us in the frame . The center of the rectangle is taken that is detecting the vehicle and when that mid portion of the rectangle crosses the yellow line, the line becomes yellow and count is incremented by 1.

III. RESULT

The experimental result consists of comparing automatic car counts from videos compared to manual calculations.. Automatic vehicle counting may be slightly less than the actual number of vehicles sometimes due to congestion and heavy traffic flow. The table shown below shows the experimental results obtained by our proposed method.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:01/January-2021 Impact Factor- 5.354 www.irjmets.com

No. of vehicles in the video(in 40 sec duration)	No. of vehicles calculated by the system	Success Rate in percent
26	25	96.15



IV. CONCLUSION

A simple and effective system which solves the traffic congestion problem under study has been developed. Vehicle detection in case of low and medium traffic is precise and the counting algorithm is accurate. The limitation of developed system is that every time when the camera feeds the data a substantial amount of tuning of parameters is required to attain the most effective performance. Also, it needs more time interval for processing in highly denced traffic conditions

V. REFERENCES

[1] T. Horprasert, D Harwood, L. S. Davis, “A statistical approach for real time robust background subtraction and shadow detection,” in proceedings of IEEE ICCV’99 Frame rate workshop, 1999, pp. 1-19.

[2] Suvarna Nandyal1, Pushpalata Patil; “Vehicle Detection and Traffic Assessment Using Images”, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 9, September 2013, pg.8 – 17.

[3] Ravi Kumar Kota, Chandra Sekhar Rao T; “Analysis Of Classification And Tracking In Vehicles Using Shape Based Features”, International Journal of Innovative Research and Development, IJIRD, Vol 2. Issue 8, pp. 279-286, August 2013.

[4] https://www.researchgate.net/publication/267272082_Vehicle_Detection_and_Tracking_Techniques_A_Concise_Review.

[5] https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9.

[6] Bin-Feng Lin; Yi-Ming Chan; Li-Chen Fu; Pei-Yung Hsiao; Li-An Chuang; Shin-Shinh Huang; Min-Fang Lo; "Integrating Appearance and Edge Features for Sedan Vehicle Detection in the Blind-Spot Area," *Intelligent Transportation Systems, IEEE Transactions on* , vol.13, no.2, pp.737,747, June 2012.

[7] Saksena MC, da Silva J, Agrawala AK (1994) Design and implementation of Maruti-II. In: Sang Son (ed.) Principles of Real-Time Systems. Prentice-Hall, Englewood Cliffs, N.J.

[8] Ikeda I, Ohnaka S, Mizoguchi M (1996) Traffic measurement with a roadside vision system – individual tracking of overlapped vehicles. In: Proceedings of the 13th International Conference on Pattern Recognition, volume 3, pp 859–864.

[9] Smith SM (1995) ASSET-2: Real-time motion segmentation and shape tracking. In: Proceedings of the International Conference on Computer Vision, Cambridge, Mass., 1995. IEEE Computer Society, Piscataway, N.J., pp 237–244.

[10] Kalinke T, Tzomakas C, von Seelen W (1998) A texture-based object detection and an adaptive model-based classification. In: Proceedings of the International Conference on Intelligent Vehicles, Stuttgart, Germany, October 1998. IEEE Industrial Electronics Society, Piscataway, N.J., pp 143–148.